The Journal of Data Mining and Knowledge Discovery manuscript No.
(will be inserted by the editor)

(Preprint: accepted for publication in

***The Journal of Data Mining and Knowledge Discovery*, 2011)**

Hellinger Distance Decision Trees are Robust and Skew-Insensitive

David A. Cieslak, T. Ryan Hoens,
Nitesh V. Chawla, W. Philip Kegelmeyer

the date of receipt and acceptance should be inserted later

Abstract Learning from imbalanced data is an important and common problem. Decision trees, supplemented with sampling techniques, have proven to be an effective way to address the imbalanced data problem. Despite their effectiveness, however, sampling methods add complexity and the need for parameter selection. To bypass these difficulties we propose a new decision tree technique called Hellinger Distance Decision Trees (HDDT) which uses Hellinger distance as the splitting criterion. We analytically and empirically demonstrate the strong skew insensitivity of Hellinger distance and its advantages over popular alternatives such as entropy (gain ratio). We apply a comprehensive empirical evaluation framework testing against commonly used sampling and ensemble methods, considering performance across 58 varied datasets. We demonstrate the superiority (using robust tests of statistical significance) of HDDT on imbalanced data, as well as its competitive performance on balanced datasets. We thereby arrive at the particularly practical conclusion that for imbalanced data it is sufficient to use Hellinger trees with bagging without any sampling methods. We provide all the datasets and software for this paper online (`http://www.nd.edu/~dial/hddt`).

1 Introduction

Decision trees are among the more popular classification methods, primarily due to their efficiency, simplicity, and interpretability. While individual trees can be limited in their expressiveness due to using only axis-parallel splits, this shortcoming can be mitigated by using an ensemble of decision trees as they have demonstrated statistically significant improvements over a single decision tree classifier [1–4]. When demonstrating the success of decision trees, however,

Corresponding Author

Address(es) of author(s) should be given

most of the work has focused on relatively balanced datasets. Exacerbating this oversight, previous work has demonstrated the innate weakness of the traditional decision tree splitting criteria (i.e., entropy and Gini) when datasets have high degrees of class imbalance [5–7] due to their sensitivity to skew.

One of the weaknesses of decision trees is dealing with imbalanced datasets. A dataset is considered "imbalanced" if one class (the majority class) vastly outnumbers the other (minority class) in the training data[1] [8]. Due to the nature of learning algorithms, class imbalance is often a major challenge as it impedes the ability of classifiers to learn the minority class concept. This is due to the fact that when learning under highly imbalanced training data, classifying all instances as negative will result in high classification accuracy.

To overcome the class imbalance problem, sampling methods have become the de facto standard for improving the performance of these decision tree algorithms [9–15]. Although successful, they add an additional — and sometimes awkward — responsibility for determining the sampling parameters. We are therefore motivated to ask: *Can we improve the performance of decision trees on highly imbalanced datasets without using sampling?*

In response to this question we previously proposed Hellinger distance as a decision tree splitting criterion to build Hellinger distance decision trees (HDDT) [7]. We compared this method, in single trees, to C4.4 (gain ratio) and CART (Gini), however since CART demonstrated consistently inferior performance to both other algorithms, we omit it here.

We would like to note that we use C4.4 [16] — unpruned, and uncollapsed C4.5 with Laplace smoothing at the leaves — as opposed to traditional C4.5; that is, C4.4 is C4.5 with slightly modified default parameters. While the decision to modify the default parameters is a popular choice in the community when applying to imbalanced data, most people still use the term C4.5 [17–19]. As we believe the term C4.4 helps disambiguate the two learning methods, we adopt the terminology and recommend its use.

The use of C4.4 instead of C4.5 is supported by prior research demonstrating that C4.4 results in improved class probability estimates [19] and is more apt for highly imbalanced datasets. In order to ensure fair comparisons, we also build uncollapsed, unpruned HDDTs with Laplace smoothing.

In this paper we extend the comparative analysis between Hellinger distance and gain ratio as decision tree splitting criteria, investigate its effectiveness in ensembles of trees, and further demonstrate the robustness of Hellinger distance to high degrees of class imbalance We also include a number of ensemble and sampling methods to arrive at a compelling conclusion: *we recommend bagged HDDTs as the preferred method for dealing with imbalanced datasets when using decision trees.* This conclusion is supported by (to the best of our knowledge) one of the most comprehensive experimental studies on decision trees for imbalanced datasets to date. This is not only in terms of datasets considered (a total of 58) but also in the techniques applied (HDDT, C4.4, two

[1] Note: By convention, the negative class is the majority class, and positive class is the minority class.

bagging variants, boosting, SMOTE, and several combinations of techniques) [7,3,20,2,21,4]. Our conclusions are supported by comparing the performance of these different classifiers using robust statistical significance tests [22,23].

In summary, the key contributions of this paper are:

1. Expand on the analysis of Hellinger distance as a decision tree splitting criterion to establish the robustness and skew insensitivity first presented in [7] (Section 2). The Hellinger Distance Decision Tree (HDDT) algorithm is presented in Section 3.
2. Empirical evaluation and analysis of the performance of HDDT and C4.4 under a comprehensive framework over a variety of measures: single trees versus ensembles, both with and without sampling. A number of different ensemble methods – bagging, boosting, majority bagging – are used. The sampling methods considered in this paper include SMOTE and under-sampling. The sampling amounts are determined via the wrapper method from [14] (Section 4).
   The analysis is broken into three parts. Part one includes only binary class imbalanced datasets. Part two includes multiple class datasets with different proportions of imbalance across the classes. In both parts we consider Area Under the ROC Curve (AUC) and F_1-measure as the performance criteria. Finally, part three is comprised of balanced datasets in order to evaluate and compare HDDT versus C4.5 for relatively balanced class distributions with standard overall accuracy as the performance measure (note that we use the original C4.5 with balanced datasets, which is a standard). A total of 58 datasets are used in this paper (as compared to only 19 binary class datasets in our prior work [7]).
3. Establish HDDT as a general decision tree algorithm broadly applicable to both imbalanced and balanced datasets, achieving statistically significantly superior performance over C4.4 for imbalanced datasets and comparable performance (neither significantly better nor worse) to C4.4 for balanced datasets. We also show that HDDTs, when used with bagging or boosting, remove the need of sampling methods, which is a big jump forward for learning decision trees for imbalanced data.

2 Hellinger Distance as a Splitting Criterion

Hellinger distance is a measure of distributional divergence [24,25] which was first applied as a decision tree splitting criterion in [7]. Let (Ω,B,ν) be a measure space [26], where P is the set of all probability measures on B that are absolutely continuous with respect to ν. Consider two probability measures $P_1, P_2 \in P$. The Bhattacharyya coefficient between P_1 and P_2 is defined as:

$$p(P_1, P_2) = \int_\Omega \sqrt{\frac{dP_1}{d\nu} \cdot \frac{dP_2}{d\nu}} d\nu. \quad (1)$$

The Hellinger distance is derived using the Bhattacharyya coefficient as:

$$h_H(P_1, P_2) = 2\left[1 - \int_\Omega \sqrt{\frac{dP_1}{d\nu} \cdot \frac{dP_2}{d\nu}} d\nu\right] = \sqrt{\int_\Omega \left(\sqrt{\frac{dP_1}{d\nu}} - \sqrt{\frac{dP_2}{d\nu}}\right)^2 d\nu}. \tag{2}$$

Within machine learning, we typically compare conditional probabilities stemming from discrete counts of data, rather than continuous functions. The information available may often be expressed as $P(Y = y|X = x)$ (which we abbreviate to $P(Y_y|X_x)$) where y is drawn from some finite set of classes like $+, -$ and x is drawn from a finite set of attribute values V such as $\{red, blue, green\}$. In the case of continuous features, a variety of splits are investigated and the set of such values becomes $\{left, right\}$. Since we are interested in evaluating over a countable rather than continuous space, we may convert the integral in Equation 2 to a summation of all values and reexpress our distributions within the context of the above conditional probability as:

$$d_H(P(Y_+), P(Y_-)) = \sqrt{\sum_{i \in V} \left(\sqrt{P(Y_+|X_i)} - \sqrt{P(Y_-|X_i)}\right)^2}. \tag{3}$$

This presents a distance which quantifies the separability of two classes of data conditioned over the full set of feature values. (As an aside, we note a strong relationship between this metric and confidence-rated boosting [27].) This lends itself as a decision tree splitting criterion with the following properties:

1. $d_H(P(Y_+), P(Y_-))$ is bounded in $[0, \sqrt{2}]$
2. $d_H(\cdot, \cdot)$ is symmetric and non-negative, i.e.,
   $d_H(P(Y_+), P(Y_-)) = d_H(P(Y_-), P(Y_+)) \geq 0$
3. squared Hellinger distance is the lower bound of KL divergence [28].

One contribution of this paper is to demonstrate the skew insensitivity of Hellinger distance (Section 2.1). As can be seen from Equations 2 and 3, class priors do not influence the Hellinger distance calculation, indicating a degree of skew insensitivity. Also, it essentially captures the divergence between the feature value distributions, given the different classes. We will further study how it manages skew in the next section.

2.1 Skew Insensitivity

In our prior work [7], we demonstrated the skew insensitivity of Hellinger distance as a decision tree splitting criterion by considering the shape of the function. In this section we will revisit these considerations, and then extend this analysis by demonstrating the effects of skew in a synthetic example.

2.1.1 Comparing isometrics

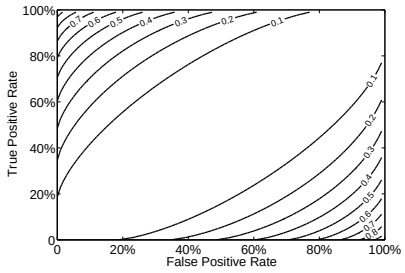
Vilalta and Oblinger [29] proposed the use of isometric lines to define the bias of an evaluation measure by plotting contours for a given measure over the range of possible values. In their paper they presented a case study on information gain, and while they did not produce isometrics under class skew, they note that "A highly skewed distribution may lead to the conclusion that two measures yield similar generalization effects, when in fact a significant difference could be detected under equal class distribution [29]." Subsequently Flach [5] connected the isometric plots to ROC analysis, demonstrating the effects of true and false positives on several common evaluation measures: accuracy, precision, and F-measure. In addition, he also presented isometrics for three major decision tree splitting criteria: entropy (used in gain ratio) [30], Gini index [31], and DKM [32]. Flach also established the effect of class skew on the shape of these isometrics [5].

This can be extended to Hellinger distance as follows:

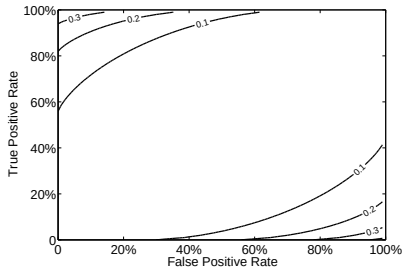
$$d_H(tpr, fpr) = \sqrt{(\sqrt{tpr} - \sqrt{fpr})^2 + (\sqrt{1 - tpr} - \sqrt{1 - fpr})^2} \qquad (4)$$

We adopt the formulation of Flach in this paper; that is, the isometric plots show the contour lines in 2D ROC space representative of the performance of different decision tree splitting criteria with respect to their estimated true and false positive rates, conditioned on the skew ratio ($c = neg/pos$). A decision tree split — for the binary class problem — can be defined by a confusion matrix as follows. A parent node will have POS positive examples and NEG negative examples. Assuming a binary split, one child will carry the true and false positive instances, and the other child will carry the true and false negative instances. The different decision tree splitting criteria, as considered in this paper, can then be modeled after this impurity (distribution of positives and negatives). Thus, in the isometric plots, each contour represents the combinations of true positives and false negatives that will generate a particular value for a given decision tree splitting criterion. For example, the 0.1 contour in Figure 2.1.1 indicates that the value of information gain[2] is 0.1 at (fpr, tpr) of approximately $(0\%, 20\%)$, $(20\%, 60\%)$, $(80\%, 100\%)$, $(20\%, 0\%)$, $(60\%, 20\%)$, $(100\%, 80\%)$, and all other combinations along the contour. In Figures 2.1.1 and 2.1.1, information gain is observed as contours formed in ROC space under a $(+ : -)$ skew of $(1 : 1)$ and $(1 : 10)$, respectively. As the skew increases, the isometrics become flatter and information gain will operate more poorly as a splitting criterion. Vilalta and Oblinger [29] and Flach [5] observed similar trends. Note that we only considered the two class proportions of $(1 : 1)$ and $(1 : 10)$ to highlight the impact of even a marginal class skew. We point the

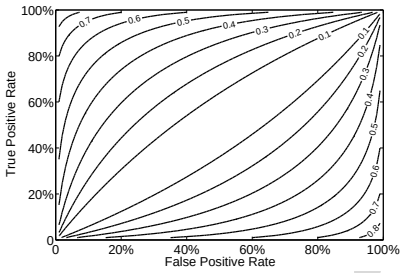
[2] Note that for these plots show information gain instead of gain ratio. The choice of information gain over gain ratio is merely for consistency with [5], however, as gain ratio and information gain are equivalent over binary splits.



(a) Information gain isometrics for an imbalance ratio of (1:1).



(b) Information gain isometrics for an imbalance ratio of (1:10).



(c) Hellinger distance isometric for any imbalance ratio

Fig. 1 Isometrics for information gain and Hellinger distance over a variety of class skews.

interested reader to the paper by Flach for a more elaborate analysis of class skew using isometrics on these three metrics [5].

Given the nature of information gain's isometric plots, we now turn our attention to Hellinger distance. First, using Flach's model of relative impurity allowed us to derive Equation 4 as an extension to Hellinger distance. In Figure 2.1.1, we see the isometric plots for Hellinger distance. While information gain showed dependence on skew in its isometric plots, we note that the Hellinger distance isometric plots do not deviate from the contours with varying class skew (c). This is due to the fact that there is no factor of c in the relative impurity formulation. The isometric contours for Hellinger distance are therefore unaffected by an increase in the class skew rate, making Hellinger distance much more robust in the presence of skew.

2.1.2 Synthetic Example

Given the analytic results from the previous section, we now wish to gain a more intuitive understanding of the potential impact of selecting between

Hellinger distance and gain ratio as a decision tree splitting criteria. Consider an artificially created dataset with two classes generated by separate Gaussian distributions of equal standard deviation with means separated by 2.5 standard deviations. In our first scenario, we simulate the effects of two equal distributions by generating 10,000 examples per class (the experiment for each class distribution is repeated over 1,000 repetitions to ensure robustness against random noise). For each repetition, we calculate the splits which are empirically chosen by C4.4, and HDDT, as well as the split which maximizes Area Under the ROC Curve (AUC) (see Section 5.2 for more details), and determine the average for each. This experiment is illustrated in Figure 2.1.2, with each vertical line indicating the average for a particular optimized split and with the error bars representing one standard deviation for each split. We note that the error bars for all three ideal splits overlap the Bayesian optimal split (where error is minimized) i.e., where the two distributions intersect. Thus, when data is balanced we expect HDDT to perform similarly to C4.4 when determining both accuracy and AUC, a result confirmed later in this report. We note that the AUC boundary is also the boundary for F-measure in this setting.

In Figure 2.1.2 we introduce a 2:1 class imbalance by sampling only 5000 points from the left distribution. We note the error bars for C4.4's split, HDDT's split, and AUC all overlap with each other, although not with that of the Bayesian optimal. This indicates that these splitting measures may not be ideal for determining accuracy, but should be theoretically optimal for AUC. We further increase class skew in Figure 2.1.2 to a ratio of 10:1. Here we begin to notice some separation between the splits of C4.4 and HDDT, as their error bars no longer overlap. The HDDT split region intersects with that of AUC, but not of accuracy. C4.4's split region, on the other hand, intersects with neither AUC nor the Bayesian optimal. Finally, we present Figure 2.1.2, which exhibits a class imbalance ratio of 100:1. C4.4 once again chooses a split region which overlaps neither the Bayesian optimal split nor the AUC split, while HDDT's split and AUC again overlap. This suggests that at levels of extreme imbalance, HDDT's can be expected to produce trees with better AUC than C4.4, and that C4.4 does not not choose ideal splits for AUC or accuracy. This conclusion is supported by observations in [7], which note that whereas the possible value continuum for C4.4 is influenced by relative class balance, the same continuum is immutable for HDDT through all possible imbalance ratios.

3 HDDT: Hellinger Distance Decision Tree

Algorithms 1 and 2 outline how Hellinger distance is incorporated into learning decision trees. We will refer to Hellinger distance and Hellinger distance based decision trees as HDDT for the remainder of the paper. In our algorithm, T_i indicates the subset of training set T which has all class i instances, $T_{x_k=j}$

specifies the subset with value j for feature k, and $T_{k,j,i}$ identifies the subset with class i and has value j for feature k.

Algorithm 1 *Calc_Binary_Hellinger*

Require: Training set T, Feature f
1: Let Hellinger $\leftarrow -1$.
2: Let V_f be the set of values of feature f.
3: **for** each value $v \in V_f$ **do**
4: Let $w \leftarrow V_f \setminus v$
5: cur_value $\leftarrow$ $(\sqrt{|T_{f,v,+}|/|T_+|} - \sqrt{|T_{f,v,-}|/|T_-|})^2 + (\sqrt{|T_{f,w,+}|/|T_+|} - \sqrt{|T_{f,w,-}|/|T_-|})^2$
6: **if** *cur_value* $>$ Hellinger **then**
7: Hellinger $\leftarrow$ *cur_value*
8: **end if**
9: **end for**
10: **return** $\sqrt{\text{Hellinger}}$

Note that Algorithm 1 is slightly different than the original definition of the Hellinger splitting criterion, in that it recommends binary splits for nominal attributes. This is due to the fact that, empirically, Hellinger distance performs better on highly branching nominal attributes with this restriction and no simple extension (similar to gain ratio vs information gain) exists. In the case that a given feature is continuous, a slight variant to Algorithm 1 is used in which *Calc_Binary_Hellinger* sorts based on the feature value, finds all meaningful splits, calculates the binary Hellinger distance at each split, and returns the highest distance; this is identical to the methodology used by C4.5 (and, by extension, C4.4). With this practical distance calculator, Algorithm 2 outlines the procedure for inducing Hellinger distance decision trees.

Algorithm 2 *HDDT*

Require: Training set T, Cut-off size C, Tree node n
1: **if** $|T| < C$ **then**
2: **return**
3: **end if**
4: $n \leftarrow argmax_f Calc\_Binary\_Hellinger(T, f)$
5: **for** each value **v** of b **do**
6: create n', a child of n
7: $HDDT(T_{x_b=v}, C, n')$
8: **end for**

Note that Algorithm 2 does not include any pruning or collapsing with Hellinger distance decision trees, and we smooth the leaf frequencies with the Laplace estimate. This was primarily motivated by the observations of Provost and Domingos [16] on C4.5.

4 Combining Sampling, Ensembles, and Decision trees

Sampling is a popular solution to the class imbalance problem; consequently a number of effective sampling methods have been proposed and studied [9–13]. We compare against two popular and effective sampling methods in this paper: random undersampling and Synthetic Minority Oversampling TEchnique (SMOTE) [13]. SMOTE and undersampling have both been shown to outperform oversampling by replication when using decision trees. In prior work [14] we demonstrated that a combination of undersampling and SMOTE generally outperforms each of the individual sampling methods as well, and proposed a wrapper method to determine the potentially optimal amounts of sampling. In the evaluations reported here, we use the same wrapper methodology to determine the amounts of sampling for both HDDT and C4.4. The wrapper discovers the sampling strategies that optimize AUC by first determining undersampling levels for majority classes in order from largest to smallest and then finding SMOTE levels for minority classes in order from smallest to largest.

In addition to these sampling methods, we evaluate multiple ensemble methods including: bagging, boosting, and majority bagging. Bagging is applied using both HDDT and C4.4 decision trees, and, to avoid any variation in results due to the choice of bootstrap replicates, we use the same bags for both HDDT and C4.4. When boosting, we use AdaBoost.M1 for binary class datasets and AdaBoost.M1W for multi-class datasets as proposed by Freund and Schapire [3]. On imbalanced datasets we also consider "majority bagging" [33] which randomly selects examples with replacement from the original training data to generate new training samples. Unlike traditional bagging, however, selection weights are assigned to ensure class balance in each new training bag. In other words, to generate each bag the majority class is undersampled and the minority class oversampled (if necessary) to generate a bag with a balanced class distribution from an imbalanced training set.

For consistency, we chose to build all ensembles with 100 decision trees (as recommended for boosting by Breiman [34]). The ensemble methods are also used with the sampling strategies. To date, ensembles have not been widely used in conjunction with sampling wrappers; hence, best practices regarding this fusion are as yet unknown. To this end, we consider multiple permutations for optimization, i.e., comparing the use of a single tree against an ensemble of trees in order to select the appropriate sampling levels.

Essentially, our experimental framework includes: single trees (T), bagging (BG), Majority Bagging (MB), AdaBoost (BT), sampling methods with parameters optimized on single trees and built with single trees, and sampling methods with ensembles of trees. We believe our work is the most extensive study with decision trees for imbalanced data to date.

Table 1 Legend of method abbreviations.

GR	Gain Ratio (C4.4)
HD	Hellinger Distance
T	Single Tree using either HD (HDDT) or GR (C4.4)
BG	Bagging
BT	Boosting
MB	Majority Bagging
SE	Balance classes with SMOTE
SW-X w/Y	Optimize sampling using classifier X, then use final classifier Y

5 Experimental Setup

In this section we outline how we compare the methods outlined previously in Section 4. Table 1 provides the abbreviations used throughout the rest of the paper.

5.1 Evaluation

In order to compare the methods, a total of 58 datasets (Tables 2 and 3) were chosen from a wide variety of application areas such as finance, biology and medicine. These datasets originate from public sources such as UCI [35], LibSVM [36], and previous studies [7,13]. In order to measure each dataset's level of imbalance, we compute the coefficient of variation (CV) which provides a measure of skew that generalizes to more than two classes [37]. Specifically, CV is the proportion of the deviation in the observed number of examples for each class versus the expected number of examples in each class. For our purposes, datasets with a CV above 0.35 — a class ratio of 2:1 on a binary dataset — are considered imbalanced. This evenly divides our pool of available datasets into 29 balanced and 29 imbalanced datasets. When evaluating each of the classifiers on the datasets, 5x2 cross-validation is used as recommended by Dietterich [23]. In this procedure, each dataset is broken into class stratified halves, allowing two experiments in which each half is once used as the training and the other in testing. This halving is iterated five times, and the average result over these ten repetitions is considered [38].

In this paper, we slightly modify the procedure from [14] when using the sampling wrapper. Each training fold is further subdivided, again using the 5x2 cross-validation methodology in order to reduce the effects of variance which may be underestimated when using 5-fold cross-validation as in the original method [23]. Each sub-training fold thus is comprised of one quarter of the original data, and the standard methodology in [14] is used to identify optimal sampling levels which are in turn applied to the original training sample to induce a final classifier evaluated on the respective testing sample.

Table 2 Statistics for the balanced datasets used in this paper.

Dataset	# Features	# Classes	# Examples	CV
breast-w	9	2	699	0.31
bupa	6	2	345	0.16
credit-a	15	2	690	0.11
crx	15	2	690	0.11
fourclass	2	2	862	0.29
heart-c	13	2	303	0.08
heart-h	13	2	294	0.28
horse-colic	22	2	368	0.26
ion	34	2	351	0.28
krkp	36	2	3196	0.04
led-24	24	10	5000	0.03
letter-26	16	26	36000	0.03
pendigits-10	16	11	10993	0.32
pima	8	2	768	0.30
promoters	57	2	106	0.00
ringnorm	20	2	300	0.09
segment-7	19	7	2310	0.00
sonar	60	2	208	0.07
splice-libsvm	60	2	1000	0.03
SVMguide1	4	2	3089	0.29
threenorm	20	2	300	0.00
tic-tac-toe	9	2	958	0.31
twonorm	20	2	300	0.01
vehicle	18	4	846	0.04
vote	16	2	435	0.23
vote1	15	2	435	0.23
vowel	10	11	528	0.00
waveform	21	3	5000	0.01
zip	256	10	9298	0.28

5.2 Evaluation Measures

In order to compare different classifiers' performance on a dataset, they must be evaluated by some evaluation measure. Typically this measure is the predictive accuracy, however this measure assumes all errors are weighted equally. This assumption is not always appropriate, e.g., when the data is imbalanced. The Receiver Operating Characteristic (ROC) curve is a standard technique for summarizing classifier performance on imbalanced datasets. Given this, a popular evaluation metric is the area under the ROC curve (AUC), which measures the probability of ranking a random positive class example over a random negative class example. We use the rank-order formulation of AUC which is akin to setting different thresholds on the probabilistic estimates and generating a *tpr* and fpr [39]. The AUC is then calculated as, given n_0 points of class 0, n_1 points of class 1, and S_0 as the sum of ranks of class 0 examples [39]: $AUC = \frac{2S_0 - n_0(n_0+1)}{2n_0n_1}$ For a multiple class dataset, we average AUC over all pairs of classes [39] using: $AUC_m = \frac{2}{c(c-1)} \sum_{i<j} AUC(i,j)$.

One advantage of AUC is that it does not rely on any threshold. This allows one to evaluate the general performance of a classifier across the different trade-offs between the *tpr* and fpr at varying decision thresholds. One disadvantage of AUC is that it does not entirely distinguish between the curves

Table 3 Statistics for the imbalanced datasets used in this paper.

Dataset	# Features	# Classes	# Examples	CV
bgp	9	4	24984	1.26
boundary	175	2	3505	0.93
breast-y	9	2	286	0.41
cam	132	2	18916	0.90
car	6	4	1728	1.08
compustat	20	2	13657	0.92
covtype	10	2	38500	0.86
credit-g	20	2	1000	0.40
dna	180	3	3186	0.39
estate	12	2	5322	0.76
germannumer	24	2	1000	0.40
glass	9	6	214	0.76
heart-v	13	2	200	0.49
hypo	25	2	3163	0.90
ism	6	2	11180	0.95
letter	16	2	20000	0.92
nursery	8	5	12961	0.95
oil	49	2	937	0.91
optdigits	64	2	5620	0.80
page	10	2	5473	0.80
page-5	10	5	5473	1.75
pendigits	16	2	10992	0.79
phoneme	5	2	5404	0.41
PhosS	480	2	11411	0.89
sat	36	6	6435	0.37
satimage	36	2	6430	0.81
segment	19	2	2310	0.71
shuttle	9	7	58000	1.87
splice	60	3	3190	0.39

that may cross in the ROC space. Thus, at a specific operating point classifier A may outperform classifier B, but the overall AUC of A may be lower. Thus choosing a classifier based on AUC may not be optimal in all cases. Under such circumstances, the problem then becomes choosing the right operating point. If one is working in a domain where the relative weights of class importance or costs of making errors are available, then the operating point can be directly chosen. Often, however, this is not the case for the datasets used in the academic literature. Hence AUC has become a popular measure of choice.

Another popular evaluation measure is F-measure. F-measure is a class of measures which captures the harmonic mean of the precision and recall of a classifier. In this paper, we consider the F_1-measure, where equal importance is given to both precision and recall. We consider the true positives (TP), false positives (FP), and false negatives (FN) as defined by a standard confusion matrix. The F_1-measure is defined as: $F_1 = \frac{2PR}{P+R}$, where $P = \frac{TP}{TP+FP}$ is precision and $R = \frac{TP}{TP+FN}$ is recall. For multiple-class imbalanced datasets, we applied a strategy similar to computing AUC over multiple classes, i.e., we average F_1-measure over all pairs of classes [39].

Finally, for balanced datasets we evaluate using the traditional accuracy measure.

5.3 Statistical Tests

Demšar [22] suggests that the best way to consider the performance of classifiers across multiple datasets is through a comparative analysis of averaged performance ranks. As previously noted, we use accuracy to rank the methods on balanced datasets, and AUC and F-measure for imbalanced datasets, where rank 1 denotes the best method. The Friedman test [40] is then performed to determine if there is a significant difference in the rankings through the Holm procedure [41], which is a step-down approach. If this procedure determines method A to rank statistically significantly ahead of method B across the considered datasets we may generally recommend the use of A over B. We note that this test requires the conservation of the sum of ranks on each dataset. Thus, in the case of a tie (scores within 0.0025) the average rank is assigned. For example, if two classifiers tie for first, they both receive a rank of 1.5, or if three tie for first, they each receive a rank of 2.

6 Imbalanced Datasets Results

For the sake of clarity, we divide the results into binary and n-ary imbalanced datasets in addition to providing a combined analysis based on the two. This differentiation is necessary as the sampling methods exhibit different performance characteristics between cases. To account for this, each minority class in the n-ary datasets will need to be considered separately to counter the problem of class imbalance.

6.1 Binary Classes

Table 4 contains the results of our experiments on binary class imbalanced datasets. The numbers reported represent the average classifier rank in terms of AUC across all the binary class imbalanced datasets for each considered method. An "×" next to a given method indicates that the method performs statistically significantly worse at that column's confidence level than the best average classifier (in the case of Table 4 that is bagged HDDT).

From Table 4 we make the following observations when using C4.4 decision trees for imbalanced data:

1. Sampling methods (SE, SW-T w/T), as expected, help C4.4 when learning on the imbalanced datasets.
2. Ensemble methods (BG, BT) are statistically significantly preferred over not only the single tree (T), but also single decision trees learned from the sampled dataset (SE, SW-T w/T). They also drive performance improvements over sampling (SW-X w/X).
3. When considering bagging (BG) in combination with the sampling wrapper (SW-X w/Y), we note that there is only a marginal separation of ranks when a single tree or ensemble of classifiers is used to optimize sampling

Table 4 AUC Ranks and statistical significance test results (at 90%, 95%, and 99% confidence levels) for binary imbalanced datasets. "×" indicates a method that performs statistically significantly worse than the best method in this table, i.e., bagged HDDT

Base Learner	Classifier	Average Rank	Confidence 90%	95%	99%
C4.4	BG	6.50			
	T	17.05	×	×	×
	BT	6.65			
	MB	10.45			
	SE	16.40	×	×	×
	SW-T w/T	16.65	×	×	×
	SW-T w/BG	8.90			
	SW-BG w/BG	8.15			
	SW-T w/BT	8.88			
	SW-BT w/BT	7.72			
HDDT	**BG**	**4.40**			
	T	14.95	×	×	×
	BT	7.22			
	MB	9.20	×		
	SE	15.40	×	×	×
	SW-T w/T	16.50	×	×	×
	SW-T w/BG	10.05	×	×	
	SW-BG w/BG	8.20			
	SW-T w/BT	8.55			
	SW-BT w/BT	8.18			

levels (SW-T versus SW-BG and SW-BT), indicating that a single tree is a sufficient heuristic for bagging in these circumstances and may be used in lieu of bagging in the optimization phase to conserve computational expense.

Based on these overall results, we recommend the use of boosting, bagging, and the sampling wrapper with boosting when using C4.4 on imbalanced datasets.

The following observations can be derived for HDDTs from Table 4:

1. A single HDDT (T) removes the need for sampling (SE, SW-T w/T). This seems a significant result, as it shows how to learn (single) decision trees for skewed data without sampling while still improving performance.
2. Ensemble methods (BG, MB, BT) significantly outperform the single HDDTs (T), as was also observed with C4.4.
3. Bagging HDDT (BG) rather than boosting (BT) is the top performer in this set of results, and has the best overall rank among all considered classifiers. In fact, bagged HDDTs are the best performing classifiers across all (including C4.4 based) classifiers. We do note that bagging and boosting both types of decision trees will typically produce favorable AUC results.

To summarize, bagging HDDT is the strongly preferred method, as it averages two ranks ahead of the next best method (the C4.4 bagging solution). This indicates that bagging will generally give the best AUC performance on imbalanced datasets with two classes and we therefore recommend the use of Hellinger distance decision trees with bagging when the class imbalance CV is above 0.35. In Section 7 we extend this result by showing that no harm is

done when using Hellinger distance decision trees with bagging when the CV is lower than 0.35.

6.1.1 Leaf Probability Estimates

To try to understand the differing impact of splitting criteria on highly imbalanced binary class data, it might be useful to examine the actual probability estimates generated by the decision tree. In this section we compare the probability estimates generated by C4.4 and HDDT, and its impact on the classifier's AUC performance.

In order to compare the methods, for each of the datasets we ran 5x2-fold cross-validation. For each dataset we then determined how many leaves predicted 1) the minority class 2) give no prediction (i.e., contain an equal number of majority and minority class instances 3) the majority class. The results of these tests can be found in Tables 5 and 6.

In the majority of the cases we see that the Hellinger trees produced, on average, more leaves than the C4.4 trees. Furthermore, while for the cases where C4.4 built deeper trees the sizes were comparable, this was not always the case when Hellinger trees were built deeper. For the cam dataset, for instance, the average C4.4 tree had 235.22 leaves, while the average Hellinger tree had 1,344 leaves. This difference shows that Hellinger trees have the potential of growing vastly deeper trees than C4.4 is able to on the same dataset. This enables Hellinger trees to find more fine-grained differences in the datasets since it can better differentiate the data, as evidenced by making more splits to further distinguish between the positive (minority) and negative (majority) class. Previous research has demonstrated that unpruned decision trees are more effective in their predictions on minority class, and also result in improved calibrated estimates [16,19].

This becomes most obvious as the imbalance becomes worse (i.e., a CV ≥ 0.80, such datasets are denoted by bold in Tables 5 and 6). In such instances, C4.4 only builds deeper trees twice (hypo and oil), and only results in higher AUC once (letter). This is very strong evidence to the effectiveness of Hellinger trees in highly imbalanced data, and their ability to pick out fine differences in instances which lead to more accurate predictions overall.

In addition to building deeper trees, Hellinger trees are also better able to create leaves which predict a class. That is, on imbalanced datasets an average C4.4 tree creates leaves with an equal number of majority and minority class instances 22.2% of the time, while Hellinger trees create such leaves only 16.0% of the time. This is significant in classification scenarios, as it means that a randomly drawn instance from the feature space is more likely to be classified by a Hellinger tree than a C4.4 tree. This observation is equally extensible to the case of only considering datasets where the CV ≥ 0.80, in which case C4.4 averages 18.1% of such leaves and Hellinger only 12.4%.

	C4.4 Leaf Distributions			
Dataset	Pred. min.	Pred. equal	Pred. maj.	Total
boundary	955 (11.7)	**3127** (**38.5**)	4050 (49.8)	8132
breast-y	1556 (14.6)	5896 (55.5)	3176 (**29.9**)	10628
cam	1576 (13.4)	3569 (**30.3**)	6616 (56.3)	11761
compustat	1611 (**29.5**)	434 (8.0)	3412 (**62.5**)	5457
covtype	2607 (35.9)	662 (9.1)	3996 (**55.0**)	7265
credit-g	5191 (24.7)	**7785** (**37.1**)	**8009** (38.2)	**20985**
estate	230 (19.4)	84 (7.1)	874 (**73.6**)	1188
germannumer	**3609** (34.8)	**1522** (**14.7**)	**5248** (**50.6**)	**10379**
heart-v	**1996** (**64.3**)	308 (9.9)	798 (25.7)	**3102**
hypo	**1826** (**63.3**)	**369** (12.8)	**690** (23.9)	**2885**
ism	1104 (**33.4**)	303 (9.2)	1901 (57.5)	3308
letter	1529 (**26.6**)	**677** (**11.8**)	3532 (61.6)	5738
oil	**428** (26.4)	**173** (**10.7**)	**1023** (**63.0**)	**1624**
page	1945 (**43.4**)	485 (10.8)	2055 (**45.8**)	4485
pendigits	**1562** (31.7)	**460** (9.3)	**2906** (**59.0**)	**4928**
phoneme	3817 (**42.8**)	683 (7.7)	4416 (**49.5**)	8916
PhosS	5415 (**33.3**)	**3175** (**19.6**)	7649 (47.1)	16239
satimage	**4722** (**35.0**)	**1595** (**11.8**)	7174 (53.2)	13491
segment	**441** (**37.6**)	**86** (**7.3**)	**647** (55.1)	**1174**
Totals	42120 (**29.7**)	31393 (**22.2**)	68172 (48.1)	141685

	HDDT Leaf Distributions			
Dataset	Pred min.	Pred equal	Pred maj.	Total
boundary	**1983** (**19.1**)	1288 (12.4)	**7138** (**68.6**)	**10409**
breast-y	**1614** (**14.7**)	**6194** (**56.4**)	**3176** (28.9)	**10984**
cam	**11793** (**17.5**)	**10275** (15.3)	**45132** (**67.2**)	**67200**
compustat	**5950** (28.5)	**2122** (**10.2**)	**12794** (61.3)	**20866**
covtype	**2983** (**36.5**)	**870** (**10.6**)	**4327** (52.9)	**8180**
credit-g	**5279** (**27.1**)	6348 (32.6)	7861 (**40.3**)	19488
estate	**9159** (**24.9**)	**6304** (**17.2**)	**21260** (57.9)	**36723**
germannumer	3445 (**37.1**)	1207 (13.0)	4639 (49.9)	9291
heart-v	1261 (52.1)	**346** (**14.3**)	**812** (**33.6**)	2419
hypo	1017 (51.7)	308 (**15.7**)	643 (**32.7**)	1968
ism	**2108** (25.5)	**1140** (**13.8**)	**5008** (**60.7**)	**8256**
letter	**1559** (26.6)	584 (9.9)	**3728** (**63.5**)	**5871**
oil	397 (**27.8**)	147 (10.3)	885 (61.9)	1429
page	**2896** (41.8)	**993** (**14.3**)	**3039** (43.9)	**6928**
pendigits	1259 (**31.9**)	404 (**10.2**)	2288 (57.9)	3951
phoneme	**11133** (41.5)	**3298** (**12.3**)	**12388** (46.2)	**26819**
PhosS	**8692** (25.1)	3118 (9.0)	**22859** (**65.9**)	**34669**
satimage	4661 (33.7)	1349 (9.7)	**7838** (**56.6**)	**13848**
segment	275 (33.3)	54 (6.5)	498 (**60.2**)	827
Totals	**77464** (26.7)	**46349** (16.0)	**166313** (**57.3**)	**290126**

Table 5 Comparing the leaves of 50 C4.4 trees and 50 Hellinger trees. For each tree type, the total number of leaves (and relative percentages) are given which predict 1) the minority class 2) give no prediction (i.e., contain an equal number of majority and minority class instances) 3) the majority class 4) the total number of leaves. Dataset names in bold indicate a CV ≥ 0.80.

6.2 Multiple Classes

Table 4 examined imbalanced data with binary classes; Table 7 repeats the analysis for imbalanced data with more than two classes. Here we note that

Dataset	C4.4	HDDT
boundary	0.57722 (2)	**0.60206 (1)**
breast-y	**0.60304 (1)**	0.58859 (2)
cam	0.64260 (2)	**0.68248 (1)**
compustat	0.81276 (2)	**0.83553 (1)**
covtype	0.97960 (2)	**0.98309 (1)**
credit-g	**0.68062 (1)**	0.68055 (2)
estate	**0.59645 (1)**	0.58821 (2)
germannumer	0.69741 (2)	**0.70887 (1)**
heart-v	**0.62668 (1)**	0.58499 (2)
hypo	0.97721 (2)	**0.98138 (1)**
ism	0.89895 (2)	**0.91360 (1)**
letter	**0.99518 (1)**	0.99214 (2)
oil	0.81574 (2)	**0.83104 (1)**
page	0.97802 (2)	**0.97877 (1)**
pendigits	0.98781 (2)	**0.99254 (1)**
phoneme	0.89706 (2)	**0.90443 (1)**
PhosS	0.60976 (2)	**0.68539 (1)**
satimage	0.90868 (2)	**0.91592 (1)**
segment	0.98473 (2)	**0.99208 (1)**
rank	1.68421	1.31579

Table 6 AUC performance results (rank in parenthesis) of the experiments performed as in Table 5. Dataset names in bold indicate a CV ≥ 0.80.

boosting C4.4 had the best rank. Though it is not statistically significantly better than bagging or boosting with HDDT, perhaps this result indicates one area of improvement for HDDT. Given that distance is defined as a separation between two distributions (i.e., classes in this case), it is not trivially extensible to multiple classes, thus creating a slight dip in the performance estimates.

6.3 Summary On All Datasets

Table 8 contains the results for all 29 imbalanced datasets combined (binary and multiple class). As the bagged ensemble and the boosted ensemble were the most competitive, we only show the results on the single tree, bagged ensemble, and the boosted ensemble. Once all the datasets and methods are combined, bagged HDDT achieves the best overall performance.

6.4 Using F_1-Measure

As stated in Section 1, we wanted to evaluate HDDTs with different popular evaluation methods to avoid possible generalization of results stemming from one measure. To this end we present the F_1-measure, which is another popular measure for evaluation on imbalanced datasets. Again due to the performance characteristics of the other methods, and in order to increase clarity of presentation, the point of comparison is largely restricted to ensembles, single trees, and the sampling wrapper with a single tree. We now investigate the

Table 7 AUC Ranks and statistical significance test results (at 90%, 95%, and 99% confidence levels) for multiple class imbalanced datasets. "×" indicates a method that performs statistically significantly worse than the best method in this table, i.e., bagged HDDT

Base Learner	Classifier	Average Rank	Confidence 90%	95%	99%
C4.4	BG	7.75			
	T	14.12	×	×	×
	BT	5.19			
	MB	7.38			
	SE	13.25	×	×	×
	SW-T w/T	16.00	×	×	×
	SW-T w/BG	10.25			
	SW-BG w/BG	10.12			
	SW-T w/BT	8.38			
	SW-BT w/BT	7.44			
HDDT	BG	7.25			
	T	16.75	×	×	×
	BT	7.00			
	MB	7.25	×		
	SE	14.88	×	×	×
	SW-T w/T	17.75	×	×	×
	SW-T w/BG	10.12	×	×	
	SW-BG w/BG	9.69			
	SW-T w/BT	10.31			
	SW-BT w/BT	9.12			

Table 8 AUC Ranks and statistical significance test results (at 90%, 95%, and 99% confidence levels) for all imbalanced datasets. "×" indicates a method that performs statistically significantly worse than the best method in this table, i.e., bagged HDDT

Base Learner	Classifier	Average Rank	Confidence 90%	95%	99%
C4.4	BG	6.23			
	T	16.21	×	×	×
	BT	6.86			
HDDT	BG	5.21			
	T	15.46	×	×	×
	BT	7.10			

questions: *Do bagged HDDTs generally outperform single HDDTs?* and: *Are HDDTs superior to C4.4 (C4.5)?*

Table 9 agree with the observations obtained via AUC, i.e., HDDT is superior to C4.4. Bagged HDDT is significantly better than a single HDDT.

Thus, based on both AUC and F_1-Measure we are able to recommend Bagged HDDTs as the preferred method when dealing with imbalanced data.

7 Balanced Datasets Results

In addition to examining the results of several methods using gain ratio and Hellinger distance based trees as base classifiers on imbalanced data, we also explore performance across a number of balanced datasets to determine if there

Table 9 F_1-Measure Ranks and statistical significance test results (at 90%, 95%, and 99% confidence levels) for all imbalanced datasets. "×" indicates a method that performs statistically significantly worse than the best method in this table, i.e., boosted C4.4

Base Learner	Classifier	Average Rank	Confidence 90%	95%	99%
C4.4	BT	7.60			
	T	12.05	×	×	×
	BG	9.18			
	SW-T w/T	12.00	×	×	×
HDDT	BT	7.92			
	T	10.55	×	×	×
	BG	8.62			
	SW-T w/T	13.30	×	×	×

Table 10 Accuracy Ranks and statistical significance test results (at 90%, 95%, and 99% confidence levels) for all balanced datasets. "×" indicates a method that performs statistically significantly worse than the best method in this table, i.e., boosted C4.5

Base Learner	Classifier	Average Rank	Confidence 90%	95%	99%
C4.5	BT	2.12			
	T	5.10	×	×	×
	BG	3.03			
HDDT	BT	2.16			
	T	5.55	×	×	×
	BG	3.03			

is the same delineation between the two splitting metrics. For the balanced data sets, we use the original C4.5 method. Our conjecture was that the differences would diminish and both gain ratio and Hellinger distance would prove to be comparable for balanced datasets. As before, results are reported as average performance ranks across all considered datasets. However, for balanced datasets we used the overall accuracy performance measure, since under these conditions it is an appropriate measure. We also greatly reduce the number of methods considered to single tree, bagging, and boosting, since the other methods are appropriate only to learning from imbalance datasets.

Table 10 shows the results for these experiments. Note that there was no statistically significant difference in performance between C4.5 and HDDT, indicating that the use of HDDT is not detrimental when applied to balanced data. Only the single tree methods are statistically significantly worse than the best ensemble method. This confirms the point (already well demonstrated for gain ratio) that ensembles generally improve accuracy over single decision trees, although this was an as yet unknown result for Hellinger distance trees.

8 Conclusion

In this paper we compared bagging, boosting, and a sampling wrapper, in addition to combinations of each method with respect to two separate splitting

criterion for decision trees: gain ratio and Hellinger distance. An experimental framework using 5x2 cross-validation compared AUC and F_1-measure performance on 29 imbalanced datasets and accuracy for 29 balanced datasets, allowing a large-scale and robust analysis of relative performances. The Holm procedure of the Friedman test was used to determine the significance of results across multiple datasets.

Based on the experiments, we make a novel and practical recommendation for learning decision trees on imbalanced data, especially binary classification data. We demonstrated that HDDTs are robust in the presence of class imbalance, and when combined with bagging they mitigate the need for sampling. This is a compelling result, as it makes bagged HDDTs particularly relevant for practitioners who don't have to then concern themselves with more expensive sampling methods. We also showed that HDDTs are not significantly worse than C4.5 for balanced datasets; thus, it is sensible to use Hellinger distance over gain ratio even on balanced datasets.

In light of the observations within this report, we claim that Hellinger distance decision trees are not only skew-insensitive as suggested in [7], but also robust in their applicability to wide variety of datasets. Thus, based on the reported findings, we recommend Hellinger distance for use in place of gain ratio in generating decision tree splits. All the datasets and software used in this paper are available via `http://www.nd.edu/~dial/hddt`.

Acknowledgments

This work was supported in part by the Arthur J. Schmitt Fellowship, NSF ECCS-0926170, and the US Department of Energy through the ASC CSEE Data Discovery Program, administered by Sandia National Laboratories, contract number: DE-AC04-76DO00789. The authors would like to thank Ken Buch for his help with Avatar Tools, in addition to the reviewers and editor assigned to refereeing this report for their helpful feedback.

References

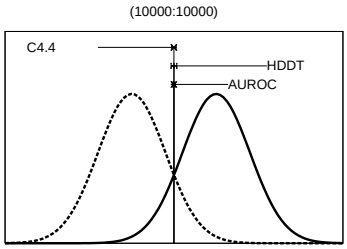
1. L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
2. ——, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
3. Y. Freund and R. Schapire, "Experiments with a New Boosting Algorithm," in *Proc. 13th Natl Conf. Machine Learning*, 1996, pp. 148–156.
4. R. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "A Comparison of Decision Tree Ensemble Creation Techniques," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 832–844, 2007.
5. P. A. Flach, "The Geometry of ROC Space: Understanding Machine Learning Metrics through ROC Isometrics," in *International Conference on Machine Learning (ICML)*, 2003, pp. 194–201.
6. C. Drummond and R. Holte, "Exploiting the cost (in)sensitivity of decision tree splitting criteria," in *International Conference on Machine Learning (ICML)*, 2000, pp. 239–246.
7. D. A. Cieslak and N. V. Chawla, "Learning Decision Trees for Unbalanced Data," in *European Conference on Machine Learning (ECML)*, 2008, pp. 241–256.

8. N. V. Chawla, N. Japkowicz, and A. Kolcz, "Editorial: Learning from Imbalanced Datasets," *SIGKDD Explorations*, vol. 6, no. 1, pp. 1–6, 2004.
9. N. Japkowicz, "Class Imbalance Problem: Significance & Strategies," in *International Conference on Artificial Intelligence (ICAI)*, 2000, pp. 111–117.
10. M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection," in *International Conference on Machine Learning (ICML)*, 1997, pp. 179–186.
11. G. Batista, R. Prati, and M. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *SIGKDD Explorations*, vol. 6, no. 1, pp. 20–29, 2004.
12. J. Van Hulse, T. Khoshgoftaar, and A. Napolitano, "Experimental Perspectives on Learning from Imbalanced Data," in *International Conference on Machine Learning (ICML)*, 2007, pp. 935–942.
13. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling TechniquE," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
14. N. V. Chawla, D. A. Cieslak, L. O. Hall, and A. Joshi, "Automatically countering imbalance and its empirical relationship to cost," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 225–252, 2008.
15. D. A. Cieslak and N. V. Chawla, "Analyzing Classifier Performance on Imbalanced Datasets when Training and Testing Distributions Differ," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2008, pp. 519–526.
16. F. Provost and P. Domingos, "Tree Induction for Probability-Based Ranking," *Machine Learning*, vol. 52, no. 3, pp. 199–215, September 2003.
17. B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers," in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 609–616.
18. C. Drummond and R. Holte, "C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling," in *ICML Workshop on Learning from Imbalanced Datasets II*, 2003.
19. N. V. Chawla, "C4.5 and Imbalanced Data Sets: Investigating the Effect of Sampling Method, Probabilistic Estimate, and Decision Tree Structure," in *ICML Workshop on Learning from Imbalanced Data Sets II*, 2003.
20. T. Ho, "The Random Subspace Method for Constructing Decision Forests," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.
21. T. Dietterich, "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization," *Machine Learning*, vol. 40, no. 2, pp. 139–157, 2000.
22. J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
23. T. G. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, 1998.
24. T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," *IEEE Transactions on Communications*, vol. 15, no. 1, pp. 52–60, February 1967.
25. C. Rao, "A Review of Canonical Coordinates and an Alternative to Correspondence Analysis using Hellinger Distance," *Questiio (Quaderns d'Estadistica i Investigacio Operativa)*, vol. 19, pp. 23–63, 1995.
26. P. Halmos, *Measure theory*. Van Nostrand and Co., 1950.
27. R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, pp. 297–336, 1999.
28. X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization," in *In Advances in Neural Information Processing Systems (NIPS*, 2007.
29. R. Vilalta and D. Oblinger, "A Quantification of Distance-Bias Between Evaluation Metrics In Classification," in *International Conference on Machine Learning (ICML)*, 2000, pp. 1087–1094.
30. J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

31. L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification and Regression Trees*. Chapman & Hall, 1984.
32. T. Dietterich, M. Kearns, and Y. Mansour, "Applying the weak learning framework to understand and improve C4.5," in *Proc. 13th International Conference on Machine Learning*. Morgan Kaufmann, 1996, pp. 96–104.
33. S. Hido and H. Kashima, "Roughly Balanced Bagging for Imbalanced Data," in *SIAM International Conference on Data Mining (SDM)*, 2008, pp. 143–152.
34. L. Breiman, "Rejoinder to the paper 'Arcing Classifiers' by Leo Breiman," *Annals of Statistics*, vol. 26, no. 2, pp. 841–849, 1998.
35. A. Asuncion and D. Newman, "UCI Machine Learning Repository," 2007. [Online]. Available: http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html
36. C. Chang and C. Lin, "LIBSVM: a library for support vector machines," 2001, software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.
37. J. Wu, H. Xiong, and J. Chen, "Cog: local decomposition for rare class analysis," *Data Mining and Knowledge Discovery*, vol. 20, pp. 191–220, 2010.
38. E. Alpaydin, "Combined 5x2cv F Test for Comparing Supervised Classification Learning Algorithms," *Neural Computation*, vol. 11, no. 8, pp. 1885–1892, 1999.
39. D. Hand and R. Till, "A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems," *Machine Learning*, vol. 45, pp. 171–186, 2001.
40. M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.
41. S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

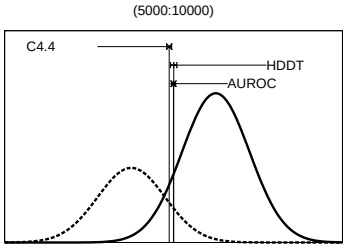
[Synthetic example with a balanced class distribution.]

(a)



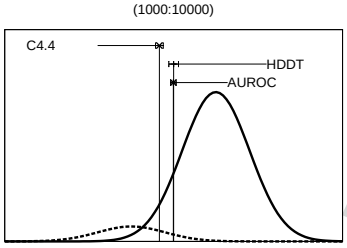
[Synthetic example with a 2:1 class distribution.]

(b)



[Synthetic example with a 10:1 class distribution.]

(c)



[Synthetic example with a 100:1 class distribution.]

(d)

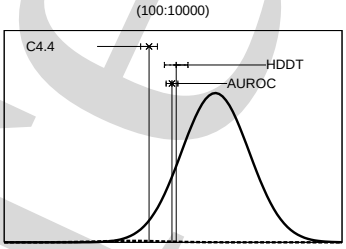


Fig. 2 Comparison of the effects of various class distributions on the ability of gain ratio and Hellinger distance to correctly determine the class boundary which optimizes AUC. Note that the Bayesian optimal split is located where the two curves intersect.